

Traveo™ Family 内蔵 CR クロック補正方法

Author: Hiroo Mizuno

Associated Part Family: Traveo Family

S6J3110/3120/3200/3310/3320/3330/3340/3350/3360/3370/3400/3510 Series

Related Documents: 一覧は[関連ドキュメント](#)を参照してください

本アプリケーションノートでは、Traveo™ family S6J3110/S6J3120/S6J3200/S6J3300/S6J3350/S6J3360/S6J3370/S6J3400/S6J3510 series に搭載される内蔵 CR クロックの補正方法を説明します。

Contents

1 はじめに	1	A S6J3110/S6J3120/S6J3200 シリーズ HCR 補正 サンプルプログラム	21
2 CR 補正機能	2	A.1 サンプルプログラム 定数および変数	21
2.1 構成	2	A.2 サンプルプログラム CR クロック周波数 測定および計算	22
2.2 ハードウェアの動作	3	A.3 サンプルプログラム CR クロック周波数補正	23
3 内蔵 CR クロック補正	4	改定履歴	29
3.1 CR クロック周波数の測定および計算	4	セールス, ソリューションおよび法律情報	30
3.2 CR クロック周波数補正	5		
4 関連ドキュメント	20		
5 まとめ	20		

1 はじめに

Traveo family S6J3110/S6J3120/S6J3200/S6J3300/S6J3350/S6J3360/S6J3370/S6J3400/S6J3510 series には内蔵高速 CR クロック, および内蔵低速 CR を搭載しています。各製品に搭載される補正可能な CR クロックを[表 1](#)にまとめます。内蔵 CR クロックの周波数はレジスタ設定により補正可能です。このレジスタに設定する補正値は、補正を行うための測定と計算より導き出せます。

表 1. 搭載される CR クロックの種類

製品	CR クロック
S6J3110/S6J3120/S6J3200/S6J3300/S6J3350	高速 CR クロック
S6J3360/S6J3370/S6J3400/S6J3510	高速 CR クロック, 低速 CR クロック

本アプリケーションノートは、内蔵 CR クロックの補正機能の使用方法について説明します。

2 CR 補正機能

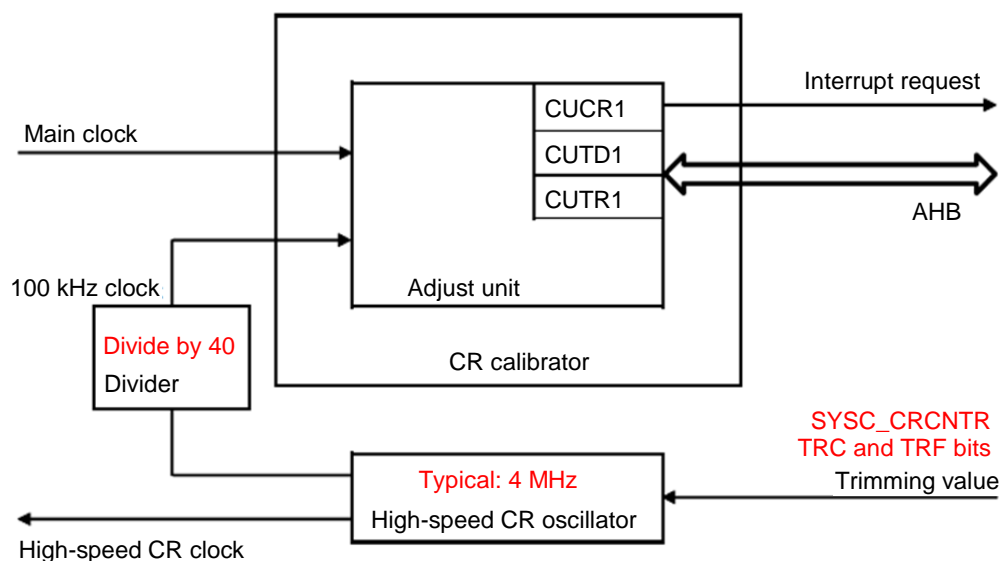
2.1 構成

図 1 に S6J3110/S6J3120/S6J3200/S6J3300/S6J3350/S6J3360/S6J3370/S6J3400/S6J3510 series のブロック図を示します。内蔵高速 CR クロックは標準 4 MHz、内蔵低速 CR クロックは標準 100 kHz です。これらはクロックシステムの CR クロック制御レジスタ(SYSC_CRCNTR)の補正値か、低速 CR クロック制御レジスタ(SYSC_SCRCNTR)によって補正されます。レジスタ SYSC_CRCNTR と SYSC_SCRCNTR は、TRC ビットと TRF ビットで構成されます。分周された内蔵高速 CR クロックは CR 補正機能の調整ユニットに入力され、メインクロックと比較されます。CR 補正機能は、この比較の完了後に割込みを生成することが可能です。

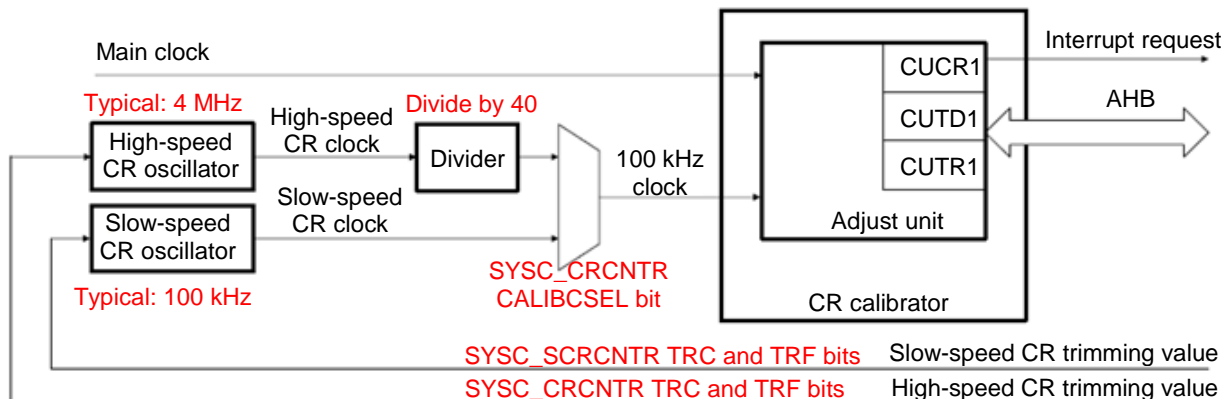
S6J3360/S6J3370/S6J3400/S6J3510 は、SYSC_CRCNTR レジスタの CALIBCSEL ビットで入力するクロックを内蔵高速 CR クロック、もしくは内蔵低速 CR クロックの、いずれかより選択可能です。

図 1. CR 補正機能 ブロック図

S6J3110/S6J3120/S6J3200/S6J3300/S6J3350 Series



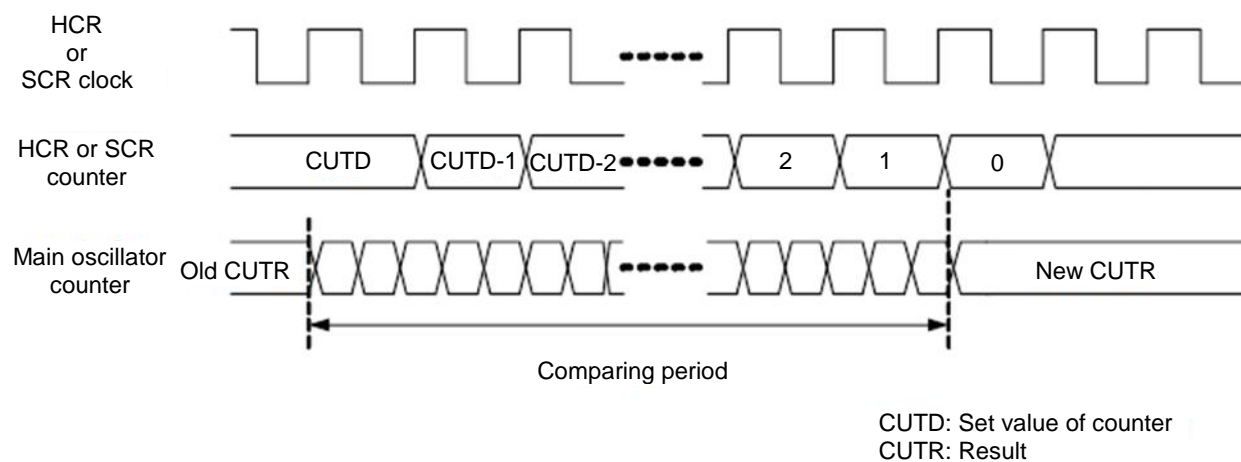
S6J3360/S6J3370/S6J3400/S6J3510 Series



2.2 ハードウェアの動作

CR 補正機能は比較期間中にメインクロックをカウントします。図 2 にタイミングチャートを示します。HCR は内蔵高速 CR クロック、また SCR は内蔵低速 CR クロックを指します。

図 2. HCR/SCR とカウンタ タイミングチャート

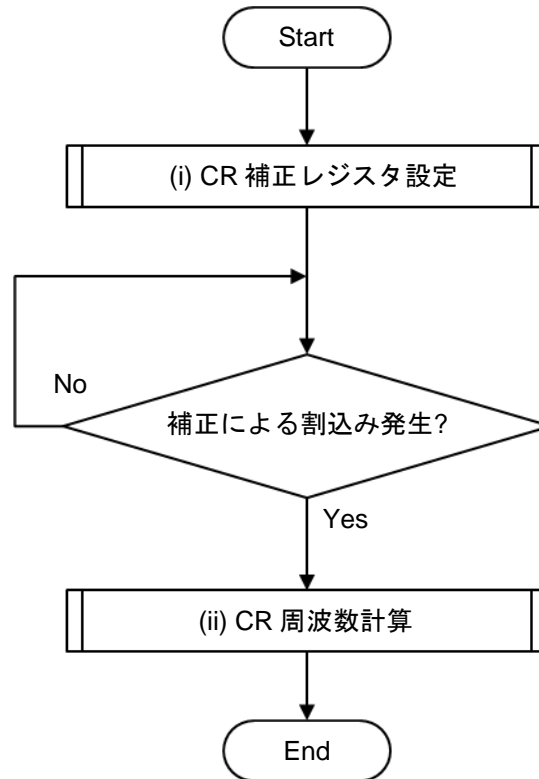


3 内蔵 CR クロック補正

3.1 CR クロック周波数の測定および計算

図 3 に HCR/SCR の周波数の測定, および計算方法のフローチャート例を示します。この手順は [CR クロック周波数補正](#) で説明する HCR/SCR の補正方法にて使用します。各手順の詳細を [図 3](#) の下に示します。

図 3. CR クロック周波数の測定/計算 フローチャート



(i) CR 補正レジスタ設定

1. CR クロックタイマデータレジスタ 1(CU_CUTD1)に値を設定します。
2. 補正機能制御レジスタ 1 の割込み許可ビット(CU_CUCR1.INTEN)に"1"を設定します。
3. 補正機能制御レジスタ 1 の補正開始ビット(CU_CUCR1.STRT)に"1"を設定します。

(ii) CR 周波数計算

1. メインクロックタイマデータレジスタ 1(CU_CUTR1)を読み取ります。
2. [式 1](#) より CR クロックを計算します。

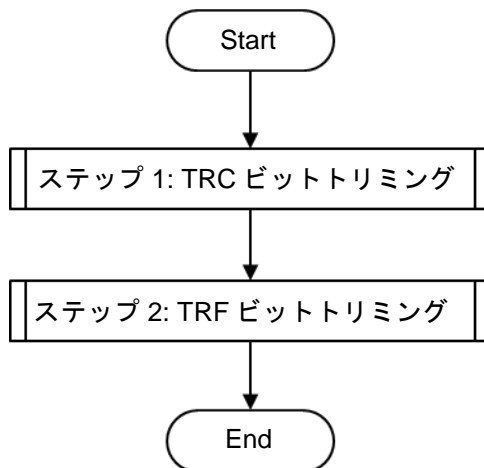
$$\text{式 1} \quad \text{CR frequency[MHz]} = \frac{\text{Main clock frequency[MHz]} \times \text{CU_CUTD1}}{\text{CU_CUTR1}}$$

3.2 CR クロック周波数補正

図 4 に HCR/SCR の補正方法のフローチャート例を示します。この手順は以下の順に行います。

1. TRC ビットによる下位分解能の補正
2. TRF ビットによる上位分解能の補正

図 4. CR クロック周波数補正 フローチャート



これらの手順は表 2 のとおり、対応製品とクロックで異なります。

表 2. 搭載される CR クロックの種類

製品およびクロック	手順 No.	参照先
S6J3110/S6J3120/S6J3200 HCR	ステップ 1 – TRC ビットトリミング	3.2.1
	ステップ 2 – TRF ビットトリミング	3.2.2
S6J3300/S6J3350/ S6J3360/S6J3370/S6J3400/S6J3510 HCR	ステップ 1 – TRC ビットトリミング	3.2.3
	ステップ 2 – TRF ビットトリミング	3.2.4
S6J3360/S6J3370/S6J3400/S6J3510 SCR	ステップ 1 – TRC ビットトリミング	3.2.5
	ステップ 2 – TRF ビットトリミング	3.2.6

3.2.1 S6J3110/S6J3120/S6J3200 HCR: ステップ 1 - TRC ビットトリミング

図 5 に TRC ビット補正のフローチャート例を示します。TRC ビットおよび TRF ビットに値を設定する前に毎回プロテクションキーのロックリリースを実施します(SYSC0_PROTKEYR="0x5CACCE55")。図 5 の詳細な手順は以下のとおりです。

1. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに 5'b11111 を設定します。
2. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRC ビットに 5'b00000 を設定します。
3. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_CA とします。
4. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRC ビットに 5'b11111 を設定します。
5. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_CB とします。
6. 式 2, 式 3, 式 4 を使用して TRC コードを計算します。この結果を C_TRC とします。

$$\text{式 2} \quad T_{CA}[\mu\text{s}] = \frac{1}{F_{CA}[\text{MHz}]}$$

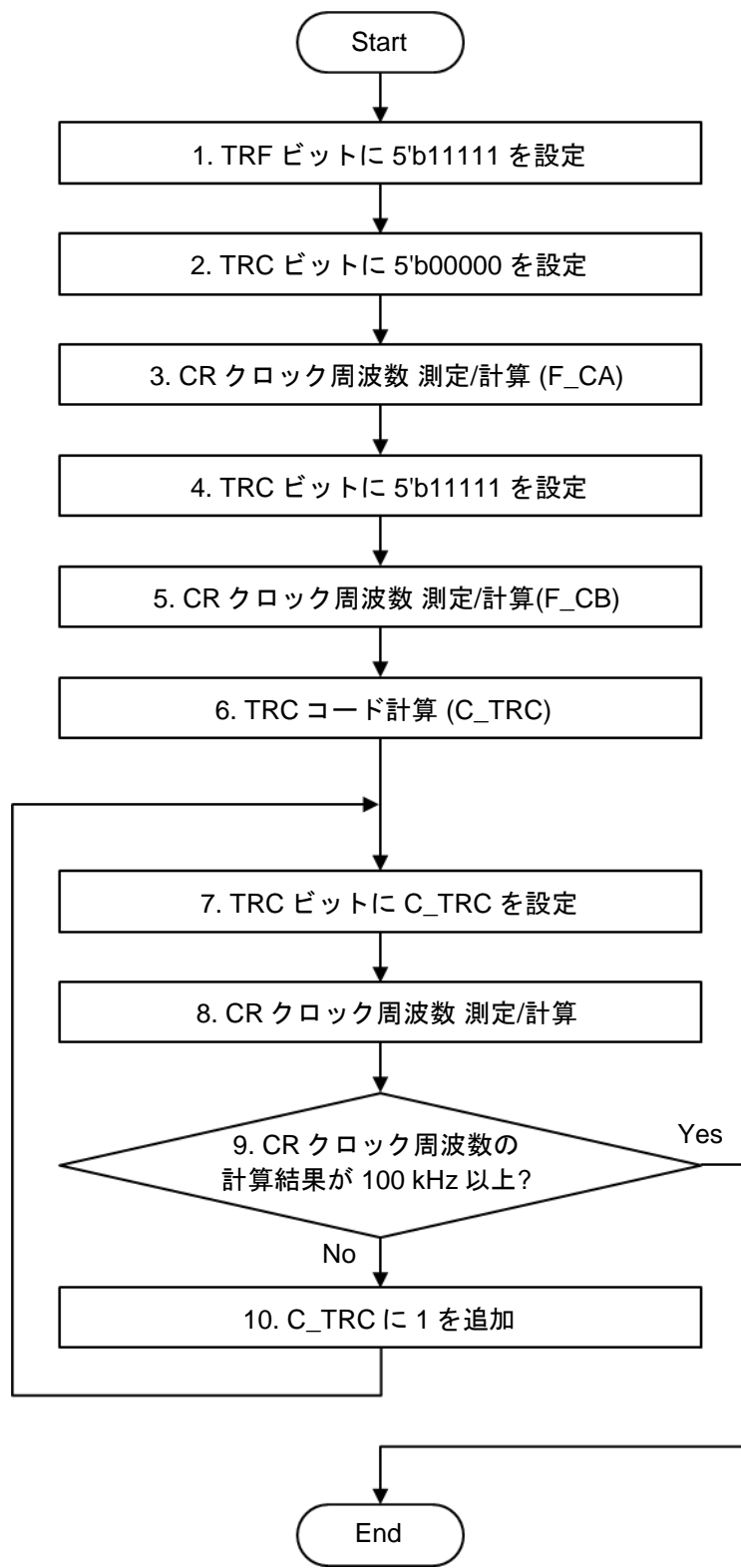
$$\text{式 3} \quad T_{CB}[\mu\text{s}] = \frac{1}{F_{CB}[\text{MHz}]}$$

$$\text{式 4} \quad C_{TRC} = \frac{(T_{CA} - 10) \times 31}{(T_{CA} - T_{CB})}$$

注意: 小数点以下は切り捨て、整数としてください。

7. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRC ビットに C_TRC を設定します。
8. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。
9. CR クロック周波数の計算結果が 100 kHz 以上の場合、S6J3110/S6J3120/S6J3200 HCR: ステップ 2 - TRF ビットトリミングに進みます。CR クロック周波数の計算結果が 100 kHz 未満の場合次の手順に進みます。
10. C_TRC に 1 を追加し、手順 7 より再度実施します。

図 5. TRC ビットトリミング フローチャート



3.2.2 S6J3110/S6J3120/S6J3200 HCR: ステップ 2 - TRF ビットトリミング

図 6 に TRF ビットの粗い補正のフローチャート例を示します。TRC ビットおよび TRF ビットに値を設定する前に、毎回プロテクションキーのロックリリースを実施します(SYSC0_PROTKEYR="0x5CACCE55")。図 6 の詳細な手順は以下のとおりです。

1. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRC ビットに S6J3110/S6J3120/S6J3200 HCR: ステップ 1 - TRC ビットトリミングで計算した C_TRC を設定します。
2. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに 5'b00000 を設定します。
3. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_FA とします。
4. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに 5'b11111 を設定します。
5. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_FB とします。
6. 式 5, 式 6, 式 7 を使用して TRF コードを計算します。この結果を C_TRF_A とします。

$$\text{式 5} \quad T_{FA}[\text{us}] = \frac{1}{F_{FA}[\text{MHz}]}$$

$$\text{式 6} \quad T_{FB}[\text{us}] = \frac{1}{F_{FB}[\text{MHz}]}$$

$$\text{式 7} \quad C_{TRF_A} = \frac{(T_{FA} - 10) \times 31}{(T_{FA} - T_{FB})}$$

注意: 小数点以下は切り捨て、整数としてください。

7. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに C_TRF_A を設定します。
8. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算 参照)を実施します。この結果を F_RA とします。
9. CR クロック周波数の計算結果が 100 kHz 以上の場合、(i) F_RA ≥ 100 kHz に進みます (図 7 を参照)。
CR クロック周波数の計算結果が 100 kHz 未満の場合、(ii) F_RA < 100 kHz に進みます (図 7 を参照)。

図 7 に TRF ビットの細かい補正のフローチャート例を示します。図 7 の(i)と(ii)の詳細な手順は以下のとおりです。

(i) F_RA ≥ 100 kHz

1. C_TRF_A より 1 を引きます。この結果を C_TRF_B とします。
2. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに C_TRF_B を設定します。
3. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_RB とします。
4. |F_RA - 100 kHz| ≤ |F_RB - 100 kHz| が成立する場合、TRF ビットに C_TRF_A を設定して完了です。
|F_RA - 100 kHz| > |F_RB - 100 kHz| が成立する場合、次の手順に進みます。
5. C_TRF_A に C_TRF_B を、F_RA に F_RB を、それぞれ代入します。その後、手順 1 より再度実施します。

(ii) F_RA < 100 kHz

1. C_TRF_A に 1 追加します。この結果を C_TRF_B とします。
2. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに C_TRF_B を設定します。
3. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_RB とします。
4. |100 kHz - F_RA| ≤ |100 kHz - F_RB| が成立する場合、TRF ビットに C_TRF_A を設定して完了です。
|100 kHz - F_RA| > |100 kHz - F_RB| が成立する場合、次の手順に進みます。
5. C_TRF_A に C_TRF_B を、F_RA に F_RB をそれぞれ代入します。その後、手順 1 より再度実施します。

図 6. TRF ビットトリミング(粗い補正) フローチャート

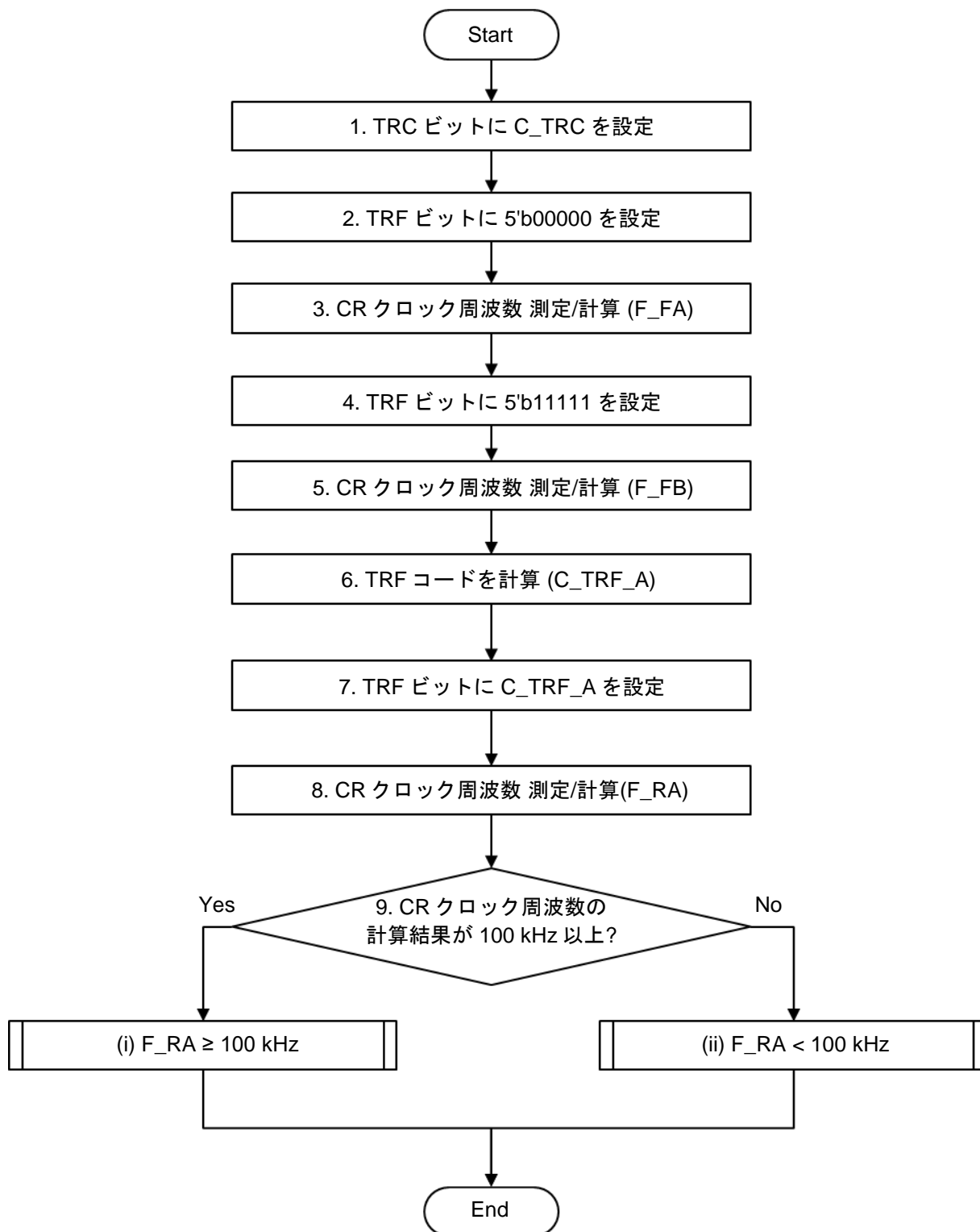
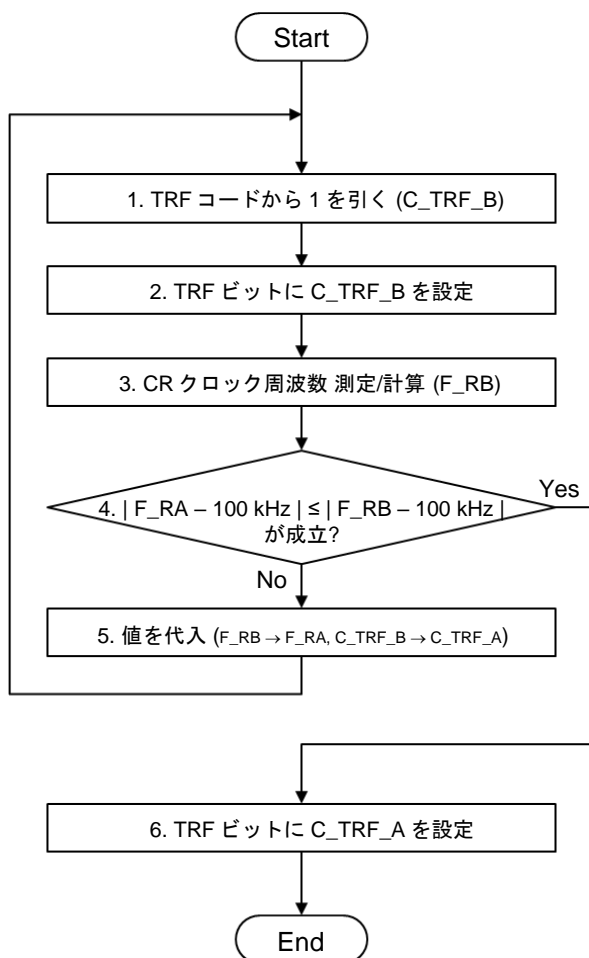
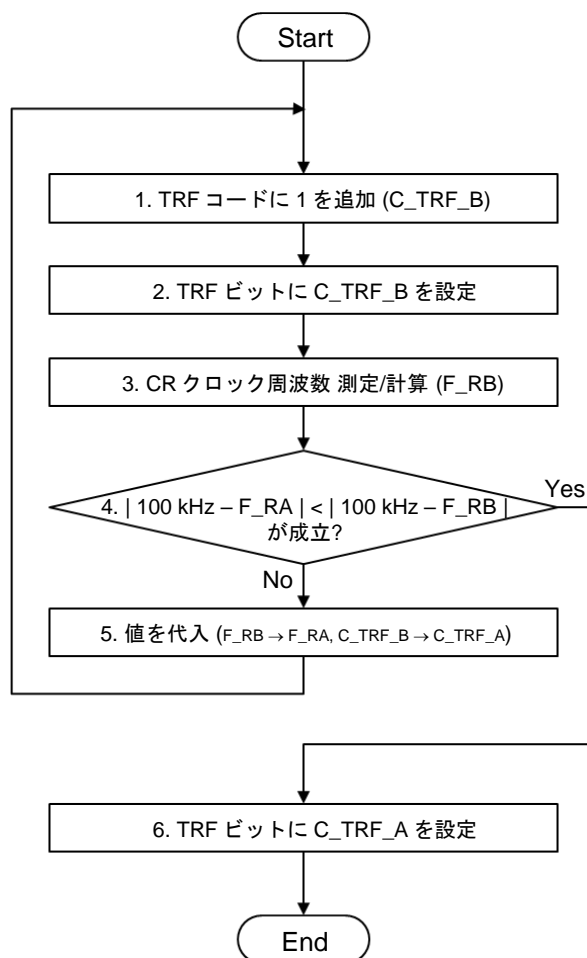


図 7. TRF ビット(細かい補正) フローチャート

(i) $F_{RA} \geq 100 \text{ kHz}$

(ii) $F_{RA} < 100 \text{ kHz}$


3.2.3 S6J3300/S6J3350/S6J3360/S6J3370/S6J3400/S6J3510 HCR: ステップ 1 - TRC ビットトリミング

図 8 に TRC ビット補正のフローチャート例を示します。TRC ビットおよび TRF ビットに値を設定する前に、毎回プロテクションキーのロックリリースを実施します(SYSC0_PROTKEYR="0x5CACCE55")。図 8 の詳細な手順は以下のとおりです。

1. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに 5'b00000 を設定します。
2. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRC ビットに 5'b10000 を設定します。
3. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_CA とします。
4. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRC ビットに 5'b10001 を設定します。
5. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_CB とします。
6. 式 8, 式 9, 式 10 を使用して TRC コードを計算します。この結果を C_TRC とします。

$$\text{式 8} \quad T_{CA}[\mu s] = \frac{1}{F_{CA}[\text{MHz}]}$$

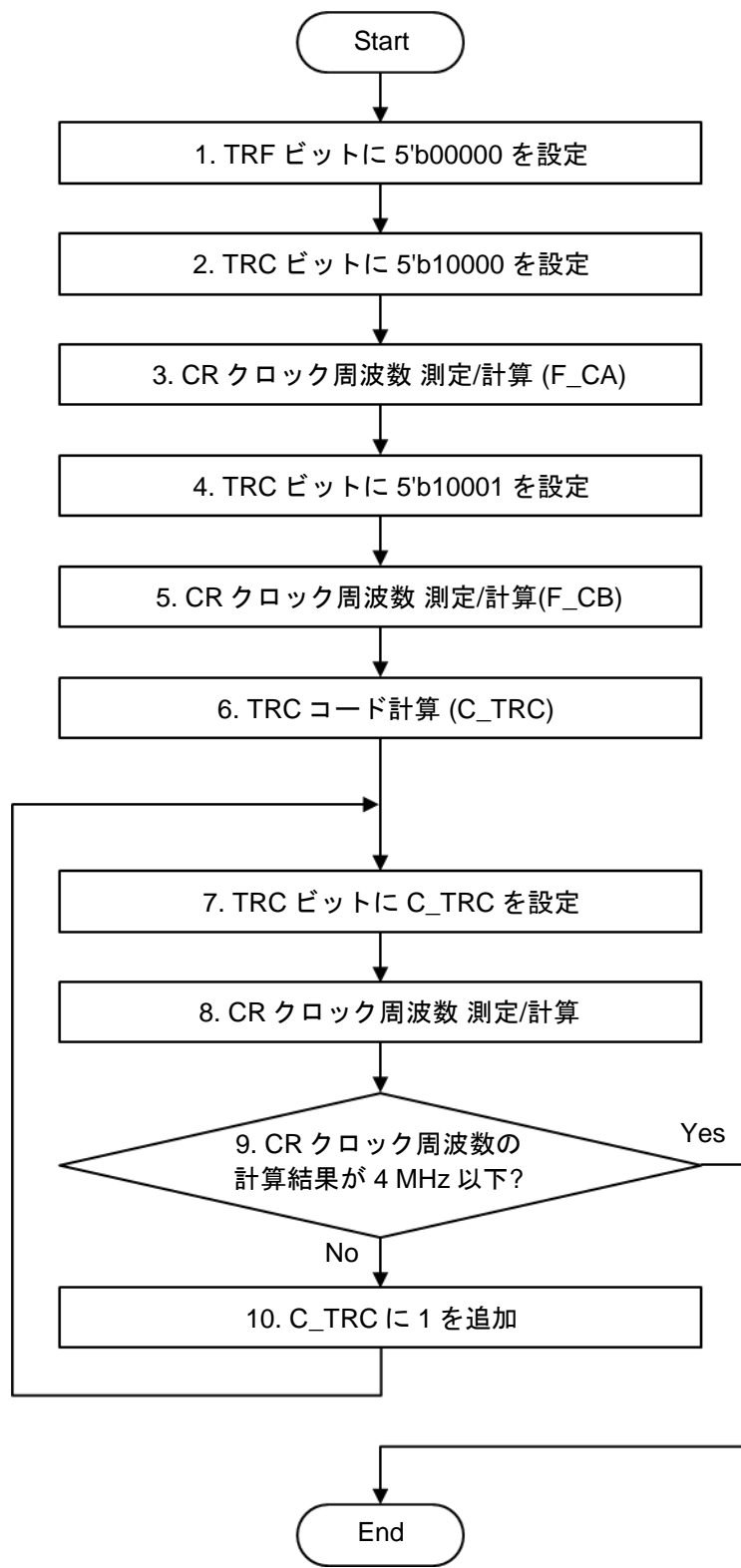
$$\text{式 9} \quad T_{CB}[\mu s] = \frac{1}{F_{CB}[\text{MHz}]}$$

$$\text{式 10} \quad C_{TRC} = \left(16 + \frac{(0.25 - T_{CA})}{(T_{CB} - T_{CA})} \right)$$

注意: 小数点以下は切り捨て、整数としてください。

7. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRC ビットに C_TRC を設定します。
8. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。
9. CR クロック周波数の計算結果が 4 MHz 以下の場合、S6J3300/S6J3350/S6J3360/S6J3370/S6J3400/S6J3510 HCR: ステップ 2 - TRF ビットトリミングに進みます。CR クロック周波数の計算結果が 4 MHz より大きい場合、次の手順に進みます。
10. C_TRC に 1 を追加し、手順 7 より再度実施します。

図 8. TRC ビットトリミング フローチャート



3.2.4 S6J3300/S6J3350/S6J3360/S6J3370/S6J3400/S6J3510 HCR: ステップ 2 - TRF ビットトリミング

図 9 に TRF ビットの補正のフローチャート例を示します。TRC ビットおよび TRF ビットに値を設定する前に、毎回プロテクションキーのロックリリースを実施します(SYSC0_PROTKEYR="0x5CACCE55")。図 9 の詳細な手順は以下のとおりです。

1. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRC ビットに S6J3300/S6J3350/S6J3360/S6J3370/S6J3400/S6J3510 HCR: ステップ 1 - TRC ビットトリミングで計算した C_TRC を設定します。
2. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに 5'b00000 を設定します。
3. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_FA とします。
4. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに 5'b11111 を設定します。
5. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_FB とします。
6. 式 11, 式 12, 式 13 を使用して TRF コードを計算します。この結果を C_TRF とします。

$$\text{式 11} \quad T_{FA}[\mu s] = \frac{1}{F_{FA}[\text{MHz}]}$$

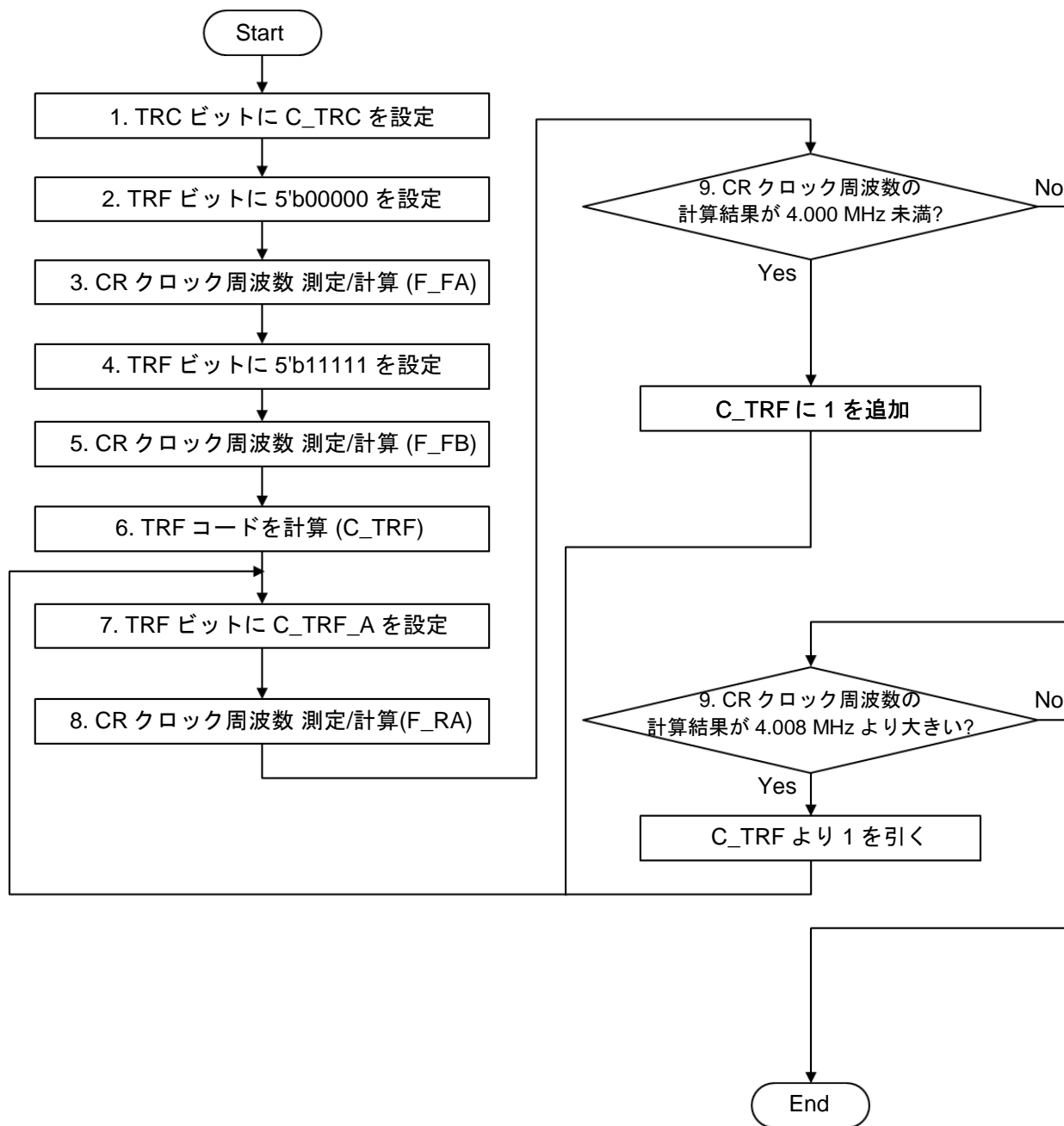
$$\text{式 12} \quad T_{FB}[\mu s] = \frac{1}{F_{FB}[\text{MHz}]}$$

$$\text{式 13} \quad C_{TRF} = \left(\frac{(0.25 - T_{FA})}{((T_{FB} - T_{FA})/31)} \right)$$

注意: 小数点以下は切り上げ、整数としてください。

7. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに C_TRF を設定します。
8. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_RA とします。
9. CR クロック周波数の計算結果が 4.000 MHz 未満の場合、C_TRF に 1 を追加し、手順 7 より再度実施します。
CR クロック周波数の計算結果が 4.008 MHz より大きい場合、C_TRF より 1 を引き、手順 7 より再度実施します。

図 9. TRF ビットトリミング フローチャート



3.2.5 S6J3360/S6J3370/S6J3400/S6J3510 SCR: ステップ 1 - TRC ビットトリミング

図 10 に TRC ビット補正のフローチャート例を示します。TRC ビットおよび TRF ビットに値を設定する前に、毎回プロテクションキーのロックリリースを実施します(SYSC0_PROTKEYR="0x5CACCE55")。図 10 の詳細な手順は以下のとおりです。

1. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに 4'b1111 を設定します。
2. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRC ビットに 3'b000 を設定します。
3. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_CA とします。
4. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRC ビットに 3'b111 を設定します。
5. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_CB とします。
6. 式 14, 式 15, 式 16 を使用して TRC コードを計算します。この結果を C_TRC とします。

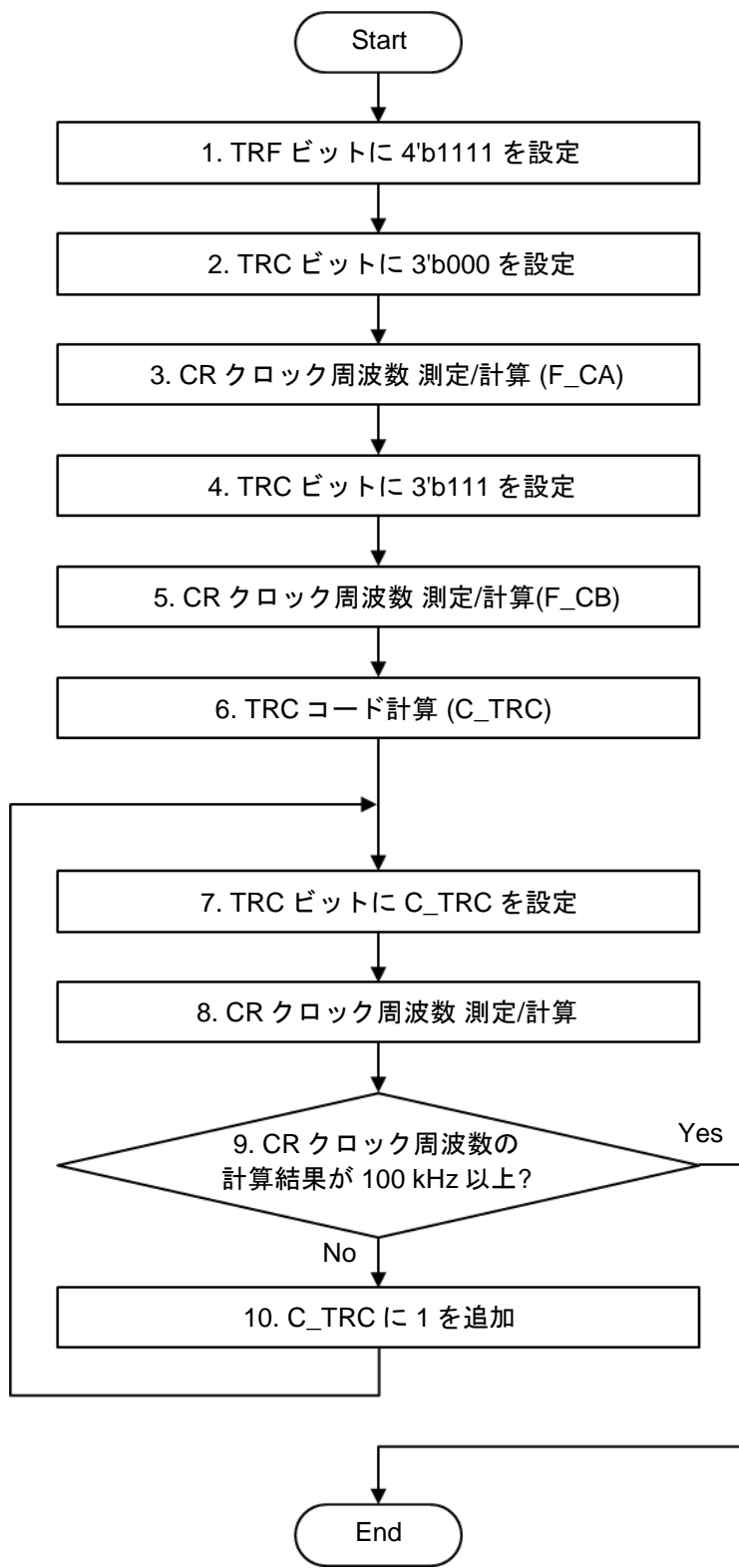
$$\text{式 14} \quad T_{CA}[\mu s] = \frac{1}{F_{CA}[\text{MHz}]}$$

$$\text{式 15} \quad T_{CB}[\mu s] = \frac{1}{F_{CB}[\text{MHz}]}$$

$$\text{式 16} \quad C_{TRC} = \left(\frac{(T_{CA} - 10)}{((T_{CA} - T_{CB})/7)} \right) \quad \text{注意: 小数点以下は切り捨て、整数としてください。}$$

7. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRC ビットに C_TRC を設定します。
8. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。
9. CR クロック周波数の計算結果が 100 kHz 以上の場合、S6J3360/S6J3370/S6J3400/S6J3510 SCR: ステップ 2 - TRF ビットトリミングに進みます。CR クロック周波数の計算結果が 100 kHz 未満の場合、次の手順に進みます。
10. C_TRC に 1 を追加し、手順 7 より再度実施します。

図 10. TRC ビットトリミング フローチャート



3.2.6 S6J3360/S6J3370/S6J3400/S6J3510 SCR: ステップ 2 - TRF ビットトリミング

図 11 に TRF ビットの粗い補正のフローチャート例を示します。TRC ビットおよび TRF ビットに値を設定する前に、毎回プロテクションキーのロックリリースを実施します(SYSC0_PROTKEYR="0x5CACCE55")。図 11 の詳細な手順は以下のとおりです。

1. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRC ビットに S6J3360/S6J3370/S6J3400/S6J3510 SCR: ステップ 1 - TRC ビットトリミングで計算した C_TRC を設定します。
2. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに 4'b0000 を設定します。
3. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_FA とします。
4. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに 4'b1111 を設定します。
5. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_FB とします。
6. 式 17, 式 18, 式 19 を使用して TRF コードを計算します。この結果を C_TRF_A とします。

$$\text{式 17} \quad T_{FA}[\mu s] = \frac{1}{F_{FA}[\text{MHz}]}$$

$$\text{式 18} \quad T_{FB}[\mu s] = \frac{1}{F_{FB}[\text{MHz}]}$$

$$\text{式 19} \quad C_{TRF_A} = \frac{(T_{FA} - 10)}{((T_{FA} - T_{FB})/15)}$$

注意: 小数点以下は切り捨て、整数としてください。

7. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに C_TRF_A を設定します。
8. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_RA とします。
9. CR クロック周波数の計算結果が 100 kHz 以上の場合、(i) F_RA ≥ 100 kHz に進みます(図 12 を参照)。CR クロック周波数の計算結果が 100 kHz 未満の場合、(ii) F_RA < 100 kHz に進みます(図 12 を参照)。

図 12 に TRF ビットの細かい補正のフローチャート例を示します。図 12 の(i)と(ii)の詳細な手順は以下のとおりです。

(i) F_RA ≥ 100 kHz

1. C_TRF_A より 1 を引きます。この結果を C_TRF_B とします。
2. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに C_TRF_B を設定します。
3. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_RB とします。
4. |F_RA - 100 kHz| ≤ |F_RB - 100 kHz| が成立する場合、TRF ビットに C_TRF_A を設定して完了です。
|F_RA - 100 kHz| > |F_RB - 100 kHz| が成立する場合、次の手順に進みます。
5. C_TRF_A に C_TRF_B を、F_RA に F_RB をそれぞれ代入します。その後、手順 1 より再度実施します。

(ii) F_RA < 100 kHz

1. C_TRF_A に 1 追加します。この結果を C_TRF_B とします。
2. プロテクションキーをロックリリース後(SYSC0_PROTKEYR="0x5CACCE55")、TRF ビットに C_TRF_B を設定します。
3. CR クロック周波数の測定および計算(CR クロック周波数の測定および計算を参照)を実施します。この結果を F_RB とします。
4. |100 kHz - F_RA| ≤ |100 kHz - F_RB| が成立する場合、TRF ビットに C_TRF_A を設定して完了です。
|100 kHz - F_RA| > |100 kHz - F_RB| が成立する場合、次の手順に進みます。

5. C_TRF_AにC_TRF_Bを、F_RAにF_RBをそれぞれ代入します。その後、手順1より再度実施します。

図 11. TRF ビットトリミング(粗い補正) フローチャート

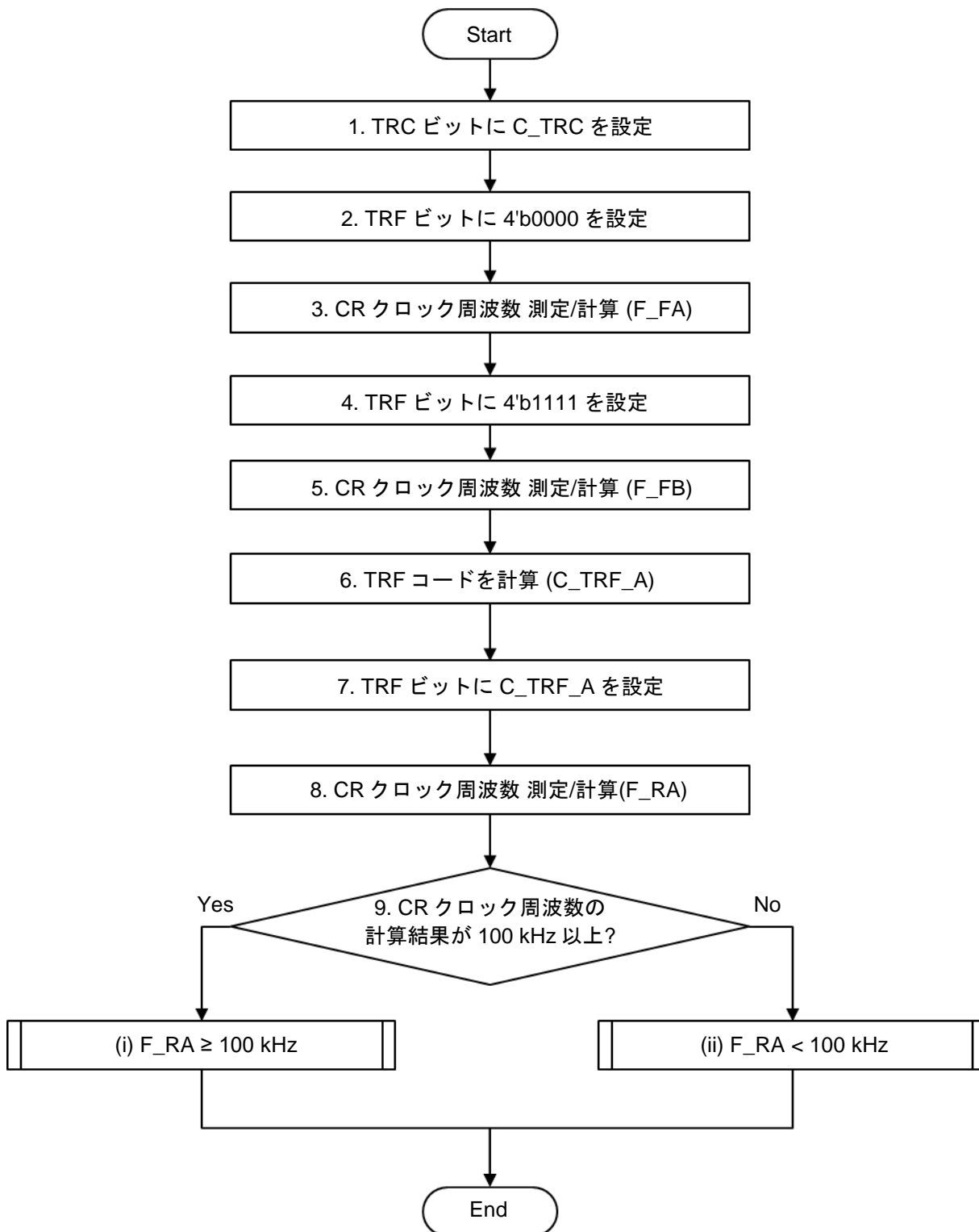
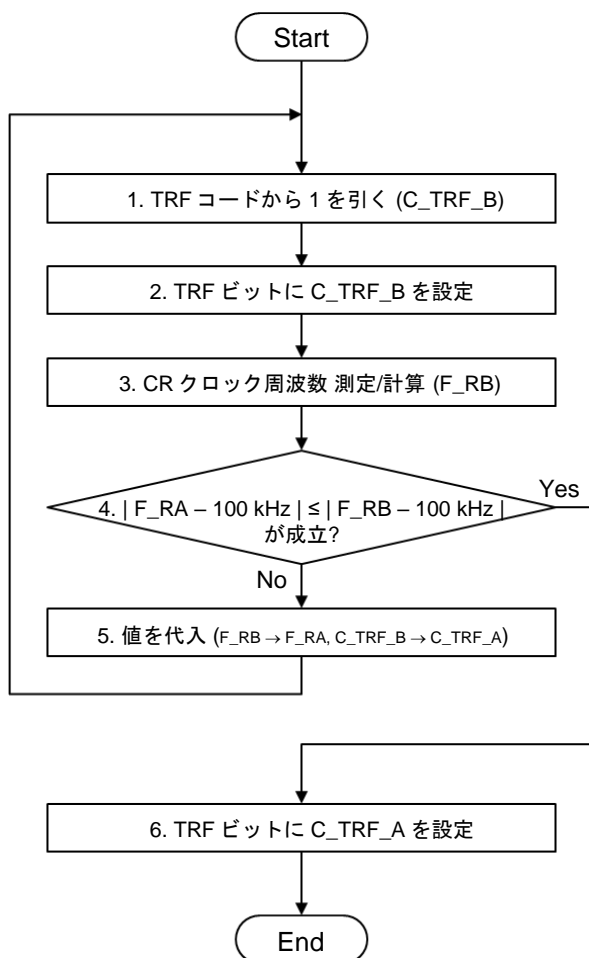
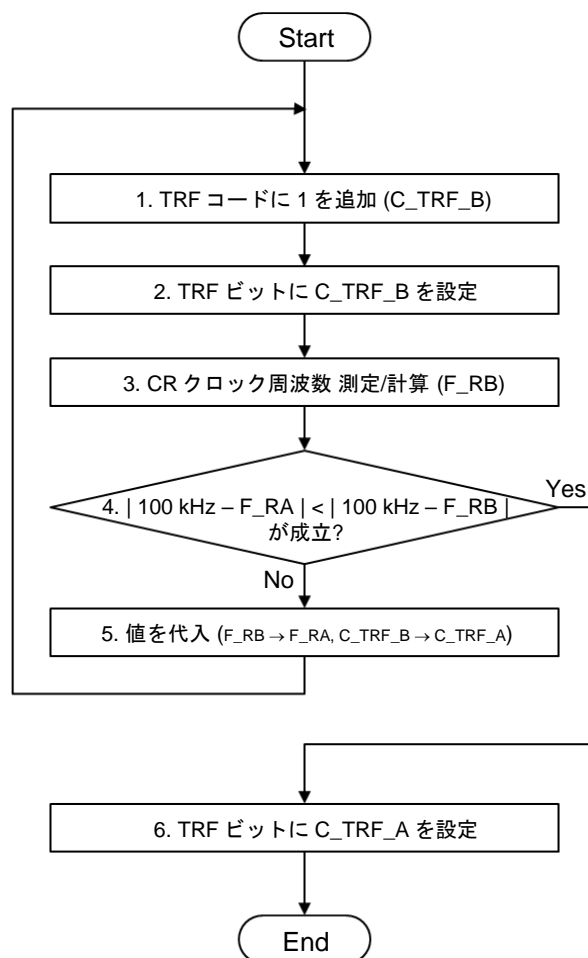


図 12. TRF ビット(細かい補正) フローチャート

 (i) $F_{RA} \geq 100 \text{ kHz}$

 (ii) $F_{RA} < 100 \text{ kHz}$


4 関連ドキュメント

Traveo ファミリのデータシートとハードウェアマニュアル:

- [S6J311E/D/C/B Series Datasheet \(Doc. No.002-05681\)](#)
- [S6J311A/9/8 Series Datasheet \(Doc. No.002-04632\)](#)
- [S6J3110 Series Hardware Manual \(Doc.No.002-10667\)](#)
- [S6J3120 Series Datasheet \(Doc.No.002-04863\)](#)
- [S6J3120 Series Hardware Manual \(Doc.No.002-04855\)](#)
- [S6J3200 Series Datasheet \(Doc.No.002-05682\)](#)
- [S6J3200 Series Hardware Manual \(Doc.No.002-04852\)](#)
- [S6J32E/F/G Series Datasheet \(Doc.No.002-10689\)](#)
- [S6J32E/F/G Series Hardware Manual \(Doc.No.002-12500\)](#)
- [Traveo Family Hardware Manual Platform Part for S6J3200 Series \(Doc.No.002-04854\)](#)
- [S6J3310/20/30/40 Series Datasheet \(Doc.No.002-10635\)](#)
- [S6J3350 Series Datasheet \(Doc.No.002-10634\)](#)
- [S6J3310/20/30/40/50 Series Hardware Manual \(Doc.No.002-10185\)](#)
- [Traveo Family Hardware Manual Platform Part for S6J3310/3320/3330/3340/3350 Series \(Doc.No.002-07884\)](#)
- [S6J3360/70 Series Datasheet \(Doc.No.002-03359\)](#)
- [S6J3360/70 Series Hardware Manual \(Doc.No.002-18302\)](#)
- [Traveo Family Hardware Manual Platform Part for S6J3360/3370 Series \(Doc.No.002-07884\)](#)
- [S6J3400 Series Datasheet \(Doc.No.001-97829\)](#)
- [S6J3400 Series Hardware Manual \(Doc.No.002-09919\)](#)
- [Traveo Family Hardware Manual Platform Part for S6J3400 Series \(Doc.No.002-07884\)](#)
- [S6J3510 Series Datasheet \(Doc.No.002-18647\)](#)
- [S6J3510 Series Hardware Manual \(Doc.No.002-18642\)](#)
- [Traveo Family Hardware Manual Platform Part for S6J3510 Series \(Doc.No.002-07884\)](#)

5 まとめ

本アプリケーションノートは、CR クロック補正機能による内蔵 CR クロック補正方法を紹介します。

A S6J3110/S6J3120/S6J3200 シリーズ HCR 補正 サンプルプログラム

この Appendix は S6J3110/S6J3120/S6J3200 シリーズの HCR 補正用のサンプルプログラムです。他の製品とクロックについては、このサンプルプログラムを参考に作成してください。

A.1 サンプルプログラム 定数および変数

コード 1 にサンプルプログラムで使用する定数および変数を示します。

コード 1. 定数および変数

```
#define TR_MAX      0x1F
#define TR_MIN      0x00 } TRC, TRF コード測定用の定数

#define UNLOCK_KEY  0x5CACCE55 SYSC_CRCNTR のロックリリース用定数

#define ABS(a, b) ((a) >= (b) ? (a-b) : (b-a)) // calculation of absolute value

#define CCM  1000000 // Calculation Coefficient 10^6
#define CCK  1000    // Calculation Coefficient 10^3

unsigned char calibration_state; // calibration procedure number

unsigned char fb_calc_mode; // selected calculation mode

unsigned long freq_calc_result;
unsigned long f_ca;
unsigned long f_cb;
unsigned long t_ca;
unsigned long t_cb;
unsigned long c_trc;
unsigned long f_tmp;
unsigned long t_tmp;
unsigned long f_fa;
unsigned long f_fb;
unsigned long t_fa;
unsigned long t_fb;
unsigned long c_trf_a;
unsigned long c_trf_b;
unsigned long f_ra;
unsigned long f_rb;
unsigned long t_ra;
unsigned long t_rb; } TRC, TRF コード測定/計算用変数

unsigned char flag_measure_complete; // software flag for CR measurement
```

A.2 サンプルプログラム CR クロック周波数 測定および計算

コード 2 に CR クロック周波数の測定および計算のサンプルプログラムを示します。これらの詳細は [CR クロック周波数の測定および計算](#) を参照してください。

コード 2. CR クロック周波数 測定および計算

図 3 (i)

図 3 (ii)

割込み関数

```

static void StartMeasureCR(void)
{
    flag_measure_complete = 0;    // reset measure complete flag (software flag)

    CU_CUTD1 = 200;                // set CR measurement duration
    CU_CUCR1 = 0x0011;            // start CR measure
}

FN_IRQ_DEFINE_BEGIN(cu_interrupt, INTERRUPTS_IRQ_NUMBER_117)
{
    unsigned long    comp_count;
    unsigned long    cr_count;

    CU_CUCRC1 = 0x0002;            // clear interrupt flag

    comp_count = CU_CUTR1;         // read main oscillation timer count
    cr_count   = CU_CUTD1;         // read CR oscillation timer count

    /* calculate frequency */
    freq_calc_result = ((4*cr_count)*CCM) / comp_count;

    flag_measure_complete = 1;    // set measure complete flag (software flag)
}
FN_IRQ_DEFINE_END()

```

任意の値

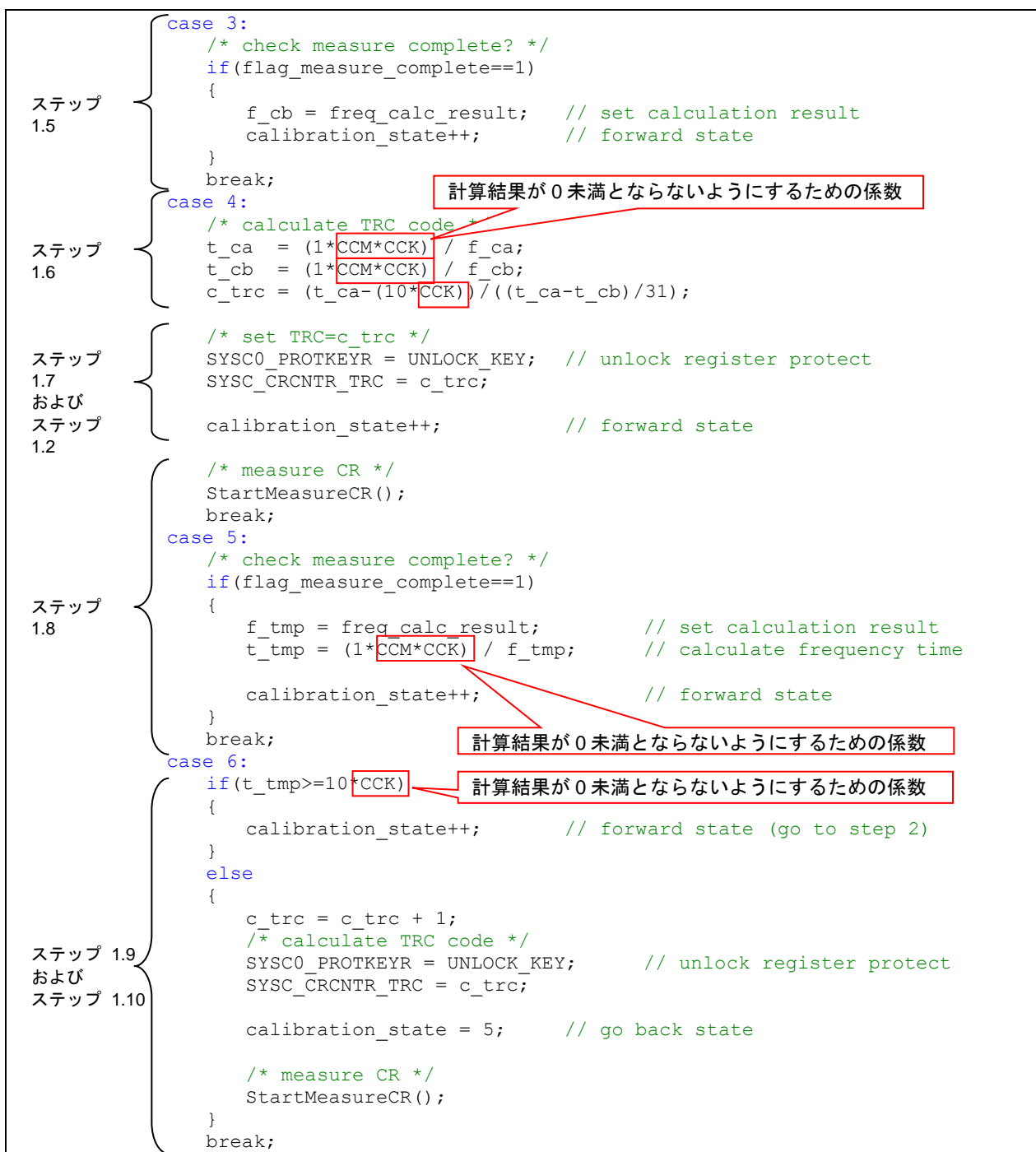
計算結果が 0 未満とならないようにするための係数

A.3 サンプルプログラム CR クロック周波数補正

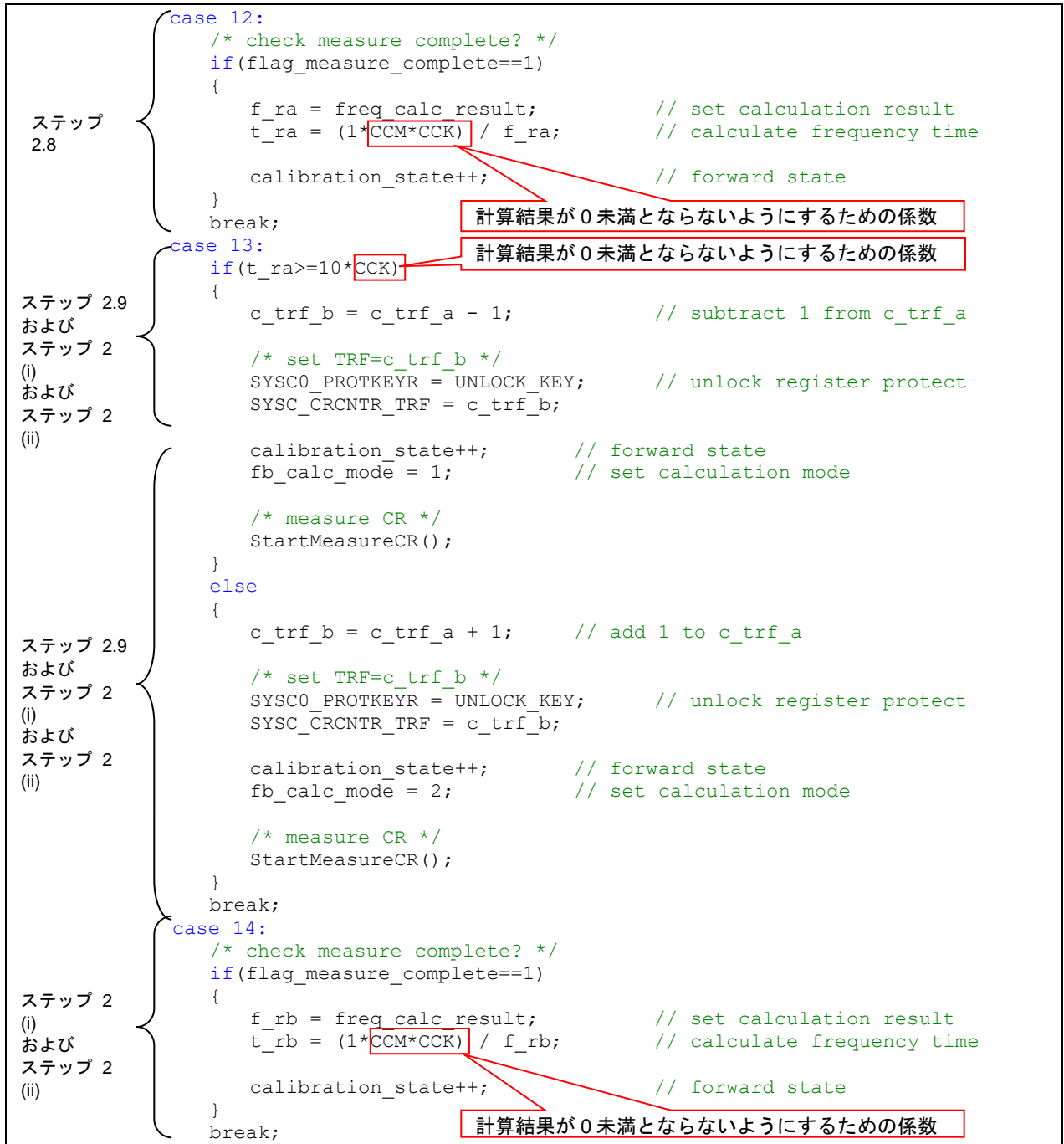
コード 3 に CR クロック周波数補正のサンプルプログラムを示します。これらの詳細は [CR クロック周波数補正](#) を参照してください。

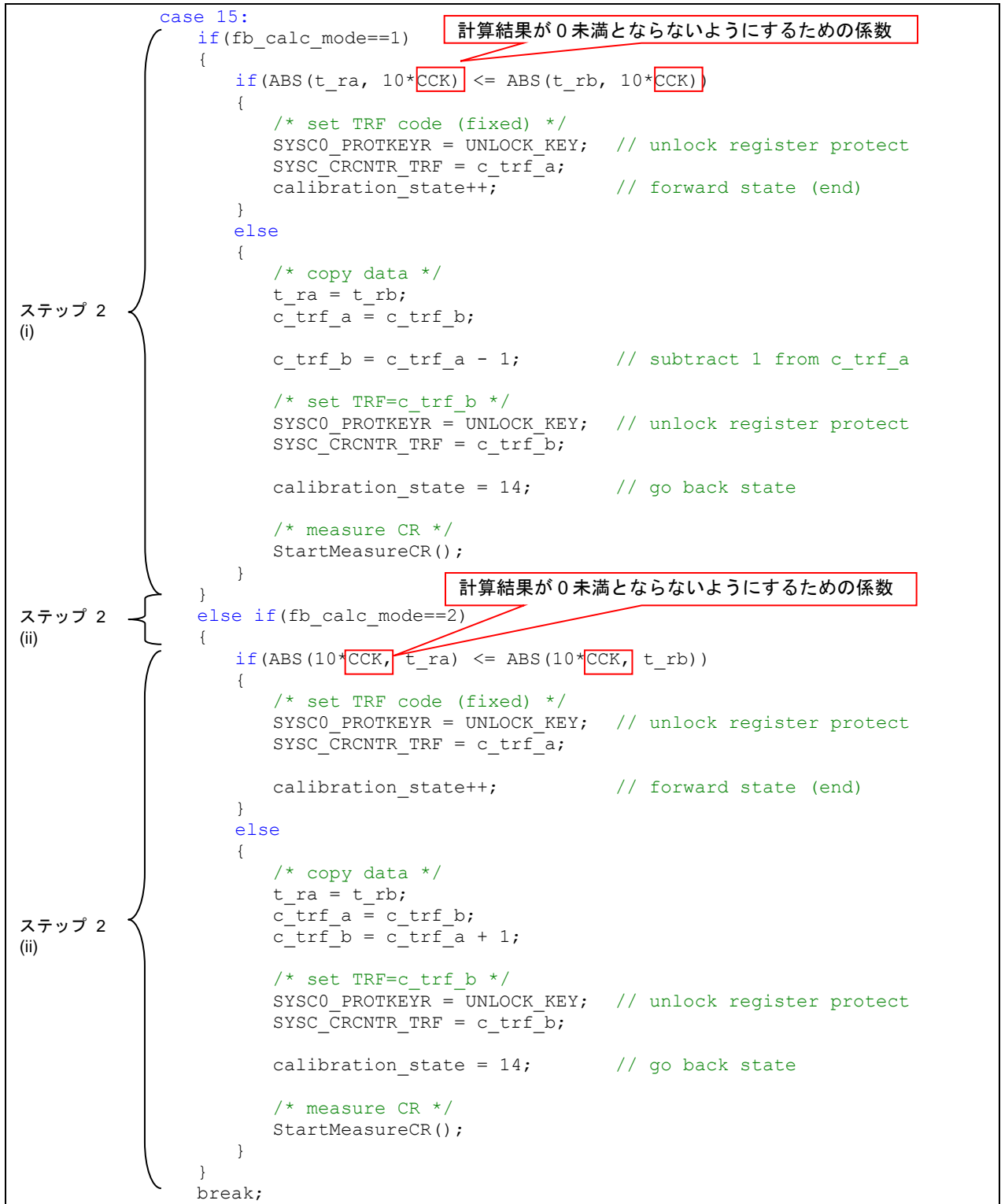
コード 3. CR クロック周波数補正

	<pre> int main(void) { // Finalize initialization to default settings. // (this will do IRQ and NMI initialization and global IRQ/NMI enable) Start_Init(); calibration_state = 0; // Endless loop for(;;) { ClearWatchdog(); switch(calibration_state) { case 0: /* ***** */ /* *** step 1 *** */ /* ***** */ /* set TRF=5'b11111 */ SYSC0_PROTKEYR = UNLOCK_KEY; // unlock register protect SYSC_CRCNTR_TRF = TR_MAX; /* set TRC=5'b00000 */ SYSC0_PROTKEYR = UNLOCK_KEY; // unlock register protect SYSC_CRCNTR_TRC = TR_MIN; calibration_state++; // forward state /* measure CR */ StartMeasureCR(); break; case 1: /* check measure complete? */ if(flag_measure_complete==1) { f_ca = freq_calc_result; // set calculation result calibration_state++; // forward state } break; case 2: /* set TRC=5'b11111 */ SYSC0_PROTKEYR = UNLOCK_KEY; // unlock register protect SYSC_CRCNTR_TRC = TR_MAX; calibration_state++; // forward state /* measure CR */ StartMeasureCR(); break; } } } </pre>
ステップ 1.1	<pre> /* ***** */ /* *** step 1 *** */ /* ***** */ /* set TRF=5'b11111 */ SYSC0_PROTKEYR = UNLOCK_KEY; // unlock register protect SYSC_CRCNTR_TRF = TR_MAX; </pre>
ステップ 1.2	<pre> /* set TRC=5'b00000 */ SYSC0_PROTKEYR = UNLOCK_KEY; // unlock register protect SYSC_CRCNTR_TRC = TR_MIN; calibration_state++; // forward state </pre>
ステップ 1.3	<pre> /* measure CR */ StartMeasureCR(); break; case 1: /* check measure complete? */ if(flag_measure_complete==1) { f_ca = freq_calc_result; // set calculation result calibration_state++; // forward state } break; </pre>
ステップ 1.4	<pre> case 2: /* set TRC=5'b11111 */ SYSC0_PROTKEYR = UNLOCK_KEY; // unlock register protect SYSC_CRCNTR_TRC = TR_MAX; calibration_state++; // forward state </pre>
ステップ 1.5	<pre> /* measure CR */ StartMeasureCR(); break; </pre>



ステップ 2.2	<pre> case 7: /***/ /*** step 2 ***/ /***/ /* set TRF=5'b00000 */ SYSC0_PROTKEYR = UNLOCK_KEY; // unlock register protect SYSC_CRCNTR_TRF = TR_MIN; calibration_state++; // forward state </pre>
ステップ 2.3	<pre> /* measure CR */ StartMeasureCR(); break; </pre>
ステップ 2.3	<pre> case 8: /* check measure complete? */ if(flag_measure_complete==1) { f_fa = freq_calc_result; // set calculation result calibration_state++; // forward state } break; </pre>
ステップ 2.4	<pre> case 9: /* set TRF=5'b11111 */ SYSC0_PROTKEYR = UNLOCK_KEY; // unlock register protect SYSC_CRCNTR_TRF = TR_MAX; calibration_state++; // forward state </pre>
ステップ 2.5	<pre> /* measure CR */ StartMeasureCR(); break; case 10: /* check measure complete? */ if(flag_measure_complete==1) { f_fb = freq_calc_result; // set calculation result calibration_state++; // forward state } break; </pre>
ステップ 2.6	<pre> case 11: /* calculate TRF code */ t_fa = (1*CCM*CCK) / f_fa; t_fb = (1*CCM*CCK) / f_fb; c_trf_a = (t_fa-(10*CCK)) / ((t_fa-t_fb)/31); </pre> <p>計算結果が0未満とならないようにするための係数</p>
ステップ 2.7	<pre> /* set TRF=c_trf_a */ SYSC0_PROTKEYR = UNLOCK_KEY; // unlock register protect SYSC_CRCNTR_TRF = c_trf_a; calibration_state++; // forward state </pre>
ステップ 2.8	<pre> /* measure CR */ StartMeasureCR(); break; </pre>





```
        case 16:
            calibration_state = 99;           // end state (calibration complete)
            break;
        default:
            break;
    }
}
```

改定履歴

Document Title: AN204096 - Traveo™ Family 内蔵 CR クロック補正方法

Document Number: 002-12931

版	ECN 番号	変更者	発行日	変更内容
**	5274266	JUMA	05/18/2016	英語版 002-04096 Rev. *A の日本語版です。
*A	5430009	JUMA	09/08/2016	英語版 002-04096 Rev. *B の日本語版です。
*B	5968792	HMIZ	11/16/2017	英語版 002-04096 Rev. *C の日本語版です。
*C	6524436	TORS	03/28/2019	英語版 002-04096 Rev. *D の日本語版です。

セールス、ソリューションおよび法律情報

ワールドワイドな販売と設計サポート

サイプレスは、事業所、ソリューション センター、メーカー代理店、および販売代理店の世界的なネットワークを保持しています。お客様の最寄りのオフィスについては、[サイプレスのロケーション ページ](#)をご覧ください。

製品

Arm® Cortex® Microcontrollers	cyress.com/arm
車載用	cyress.com/automotive
クロック&バッファ	cyress.com/clocks
インターフェース	cyress.com/interface
IoT(モノのインターネット)	cyress.com/iot
メモリ	cyress.com/memory
マイクロコントローラ	cyress.com/mcu
PSoC	cyress.com/psoc
電源用 IC	cyress.com/pmic
タッチ センシング	cyress.com/touch
USB コントローラー	cyress.com/usb
ワイヤレス	cyress.com/wireless

PSoC® ソリューション

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6 MCU](#)

サイプレス開発者コミュニティ

[コミュニティ](#) | [Projects](#) | [ビデオ](#) | [ブログ](#) | [トレーニング Components](#)

テクニカルサポート

cyress.com/support

Arm and Cortex are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2015-2019. 本書面は、Cypress Semiconductor Corporation 及び Spansion LLC を含むその子会社 (以下「Cypress」という。) に帰属する財産である。本書面 (本書面に含まれ又は言及されているあらゆるソフトウェア若しくはファームウェア (以下「本ソフトウェア」という。)) を含む) は、アメリカ合衆国及び世界のその他の国における知的財産法令及び条約に基づき Cypress が所有する。Cypress はこれらの法令及び条約に基づく全ての権利を留保し、本段落で特に記載されているものを除き、その特許権、著作権、商標権又はその他の知的財産権のライセンスを一切許諾しない。本ソフトウェアにライセンス契約書が伴っておらず、かつ Cypress との間で別途本ソフトウェアの使用方法を定める書面による合意がない場合、Cypress は、(1) 本ソフトウェアの著作権に基づき、(a) ソースコード形式で提供されている本ソフトウェアについて、Cypress ハードウェア製品と共に用いるためにのみ、かつ組織内部でのみ、本ソフトウェアの修正及び複製を行うこと、並びに (b) Cypress のハードウェア製品ユニットに用いるためにのみ、(直接又は再販売者及び販売代理店を介して間接のいずれかで) 本ソフトウェアをバイナリーコード形式で外部エンドユーザーに配布すること、並びに (2) 本ソフトウェア (Cypress により提供され、修正がなされていないもの) が抵触する Cypress の特許権のクレームに基づき、Cypress ハードウェア製品と共に用いるためにのみ、本ソフトウェアの作成、利用、配布及び輸入を行うことについての非独占的で譲渡不能な一身専属的ライセンス (サブライセンスの権利を除く) を付与する。本ソフトウェアのその他の使用、複製、修正、変換又はコンパイルを禁止する。

適用される法律により許される範囲内で、Cypress は、本書面又はいかなる本ソフトウェア若しくはこれに伴うハードウェアに関しても、明示又は黙示をとわず、いかなる保証 (商品性及び特定の目的への適合性の黙示の保証を含むがこれらに限られない) も行わない。いかなるコンピューティングデバイスも絶対に安全ということはない。従って、Cypress のハードウェアまたはソフトウェア製品に講じられたセキュリティ対策にもかかわらず、Cypress は、Cypress 製品への権限のないアクセスまたは使用といったセキュリティ違反から生じる一切の責任を負わない。加えて、本書面に記載された製品には、エラーと呼ばれる設計上の欠陥またはエラーが含まれている可能性があり、公表された仕様とは異なる動作をする場合がある。適用される法律により許される範囲内で、Cypress は、別途通知することなく、本書面を変更する権利を留保する。Cypress は、本書面に記載のある、いかなる製品若しくは回路の適用又は使用から生じる一切の責任を負わない。本書面で提供されたあらゆる情報 (あらゆるサンプルデザイン情報又はプログラムコードを含む) は、参照目的のためのみに提供されたものである。この情報で構成するあらゆるアプリケーション及びその結果としてのあらゆる製品の機能性及び安全性を適切に設計、プログラム、かつテストすることは、本書面のユーザーの責任において行われるものとする。Cypress 製品は、兵器、兵器システム、原子力施設、生命維持装置若しくは生命維持システム、蘇生用の設備及び外科的移植を含むその他の医療機器若しくは医療システム、汚染管理若しくは有害物質管理の運用のために設計され若しくは意図されたシステムの重要な構成部分としての使用、又は装置若しくはシステムの不具合が人身傷害、死亡若しくは物的損害を生じさせるようなその他の使用 (以下「本目的外使用」という。) のためには設計、意図又は承認されていない。重要な構成部分とは、その不具合が装置若しくはシステムの不具合を生じさせるか又はその安全性若しくは実効性に影響すると合理的に予想できるような装置若しくはシステムのあらゆる構成部分という。Cypress 製品のあらゆる本目的外使用から生じ、若しくは本目的外使用に関連するいかなる請求、損害又はその他の責任についても、Cypress はその全部又は一部をとわず一切の責任を負わず、かつ Cypress はそれら一切から本書により免除される。Cypress は Cypress 製品の本来目的外使用から生じ又は本目的外使用に関連するあらゆる請求、費用、損害及びその他の責任 (人身傷害又は死亡に基づく請求を含む) から免責補償される。

Cypress、Cypress のロゴ、Spansion、Spansion のロゴ及びこれらの組み合わせ、WICED、PSoC、CapSense、EZ-USB、F-RAM、及び Traveo は、米国及びその他の国における Cypress の商標又は登録商標である。Cypress のより完全な商標のリストは、cyress.com を参照すること。その他の名称及びブランドは、それぞれの権利者の財産として権利主張がなされている可能性がある。